Think Like a Hacker

@Brunty

This is not a talk about in-depth, detailed exploit techniques

@Brunty

Developer

Mentor & mentee

Tinkerer

Who are hackers?

@Brunty

> “
>
> Black hat: hacker doing evil
>
> White hat: hacker doing good
>
> Grey hat: hacker hacking
>
> Top hat: hacker doing fancy stuff

@beerbikesbacon

https://twitter.com/beerbikesbacon/status/1186783818272952327

Clever

Creative

Curious

Why do they do it?

@Brunty

Financial gain

Reputation

Corporate reasons

Ideological reasons

Stumbled upon something

What makes you a target?

Popularity

Politics & perspective

People

Pot-luck

Quick wins

What can you do to start reducing risk?

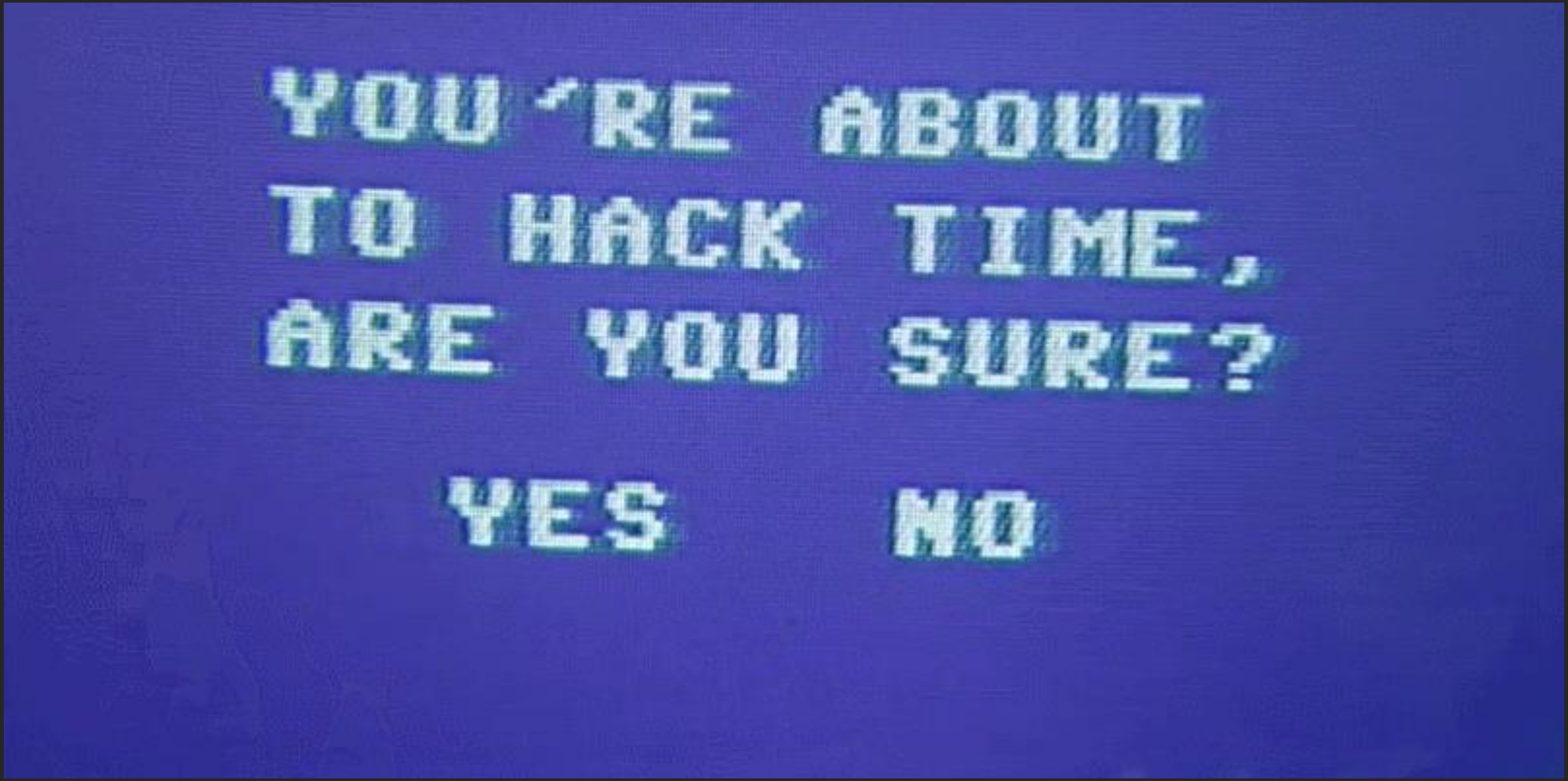
No magic solution

Embed security considerations into the whole project workflow

> No-one has the time or money for securing their systems until it’s too late

Clinton Ingrams

https://twitter.com/cfing99

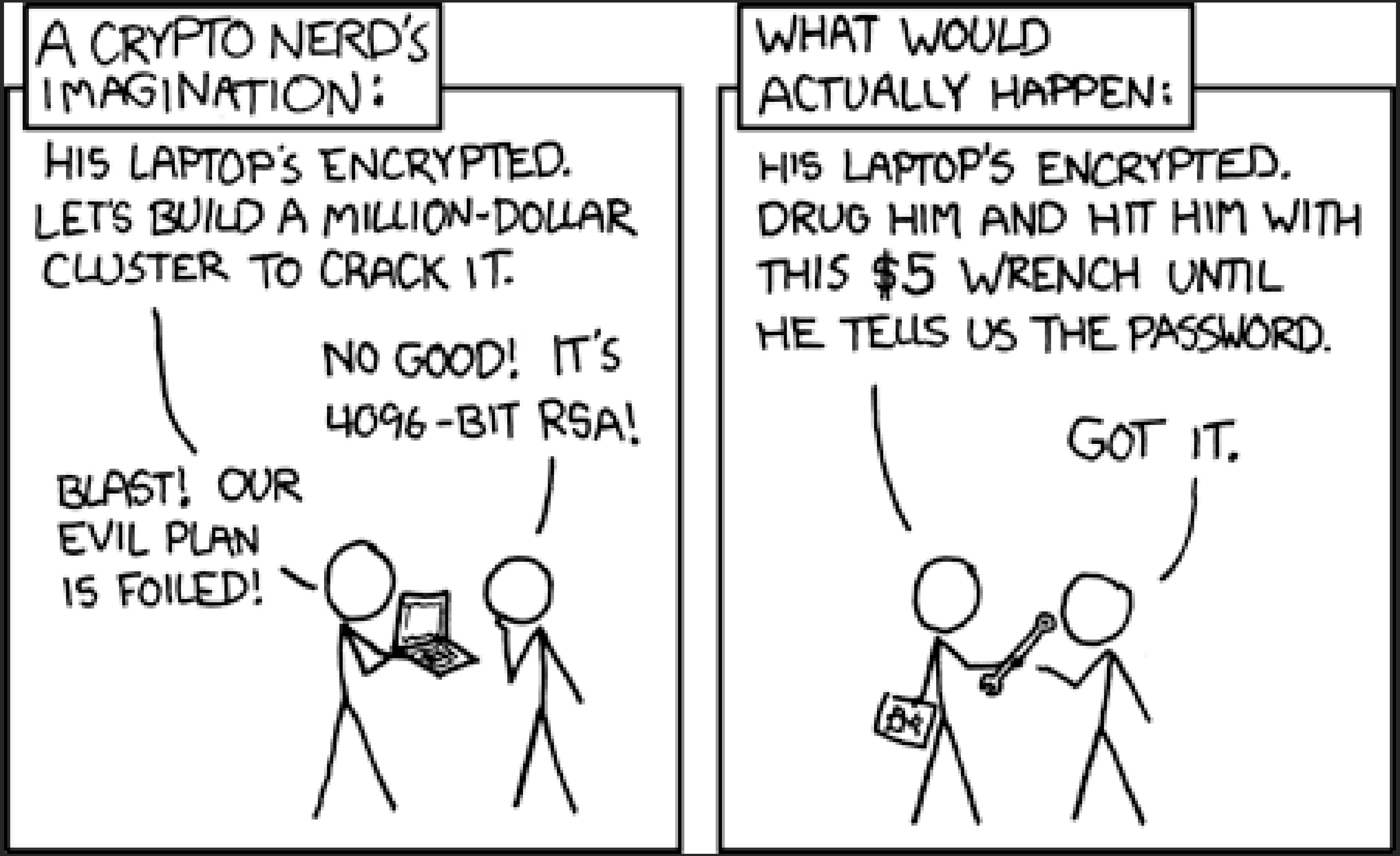


YOU'RE ABOUT
TO HACK TIME,
ARE YOU SURE?

YES NO

It is every developer's responsibility

The people problem


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

https://xkcd.com/538/

@Brunty

Principle of least privilege

Limit who has access to what

Do all your devs really need 24/7 access to your production DB?

“

> No developer should ever have a permanent login, or access to any credentials

David McKay

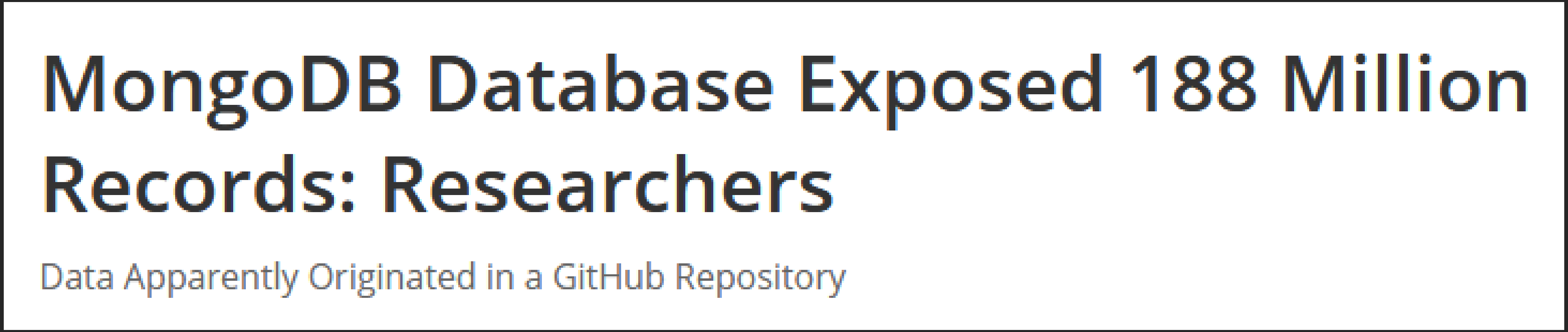
https://twitter.com/rawkode/status/1182213985661308928

> That's not to say that a “Break Glass” button in the admin interface can't generate a prod database login that's valid for an hour; but it needs to log who requested it and take a reason; and notify slack, et al

David McKay

https://twitter.com/rawkode/status/1182213789686620160

Where is your data stored?



MongoDB Database Exposed 188 Million Records: Researchers

Data Apparently Originated in a GitHub Repository

https://www.bankinfosecurity.com/mongodb-database-exposed-188-million-records-researchers-a-12769

Who are the third parties you trust with your data?

@Brunty

Who are the third parties you trust with your customer data?

@Brunty

Shodan

https://www.shodan.io

Check your repos for secrets

zricethezav/gitleaks

https://github.com/zricethezav/gitleaks

@Brunty

Check your public sites for secrets

Google dorking

DB_PASSWORD filetype:env

OSINT

https://osintframework.com

Curiosity

"what if…"

Don’t trust user input

@Brunty

“I’d like to be removed from the mailing list please”

“I’d like to be removed from the mailing list please”

Use prepared statements

https://en.wikipedia.org/wiki/Prepared_statement

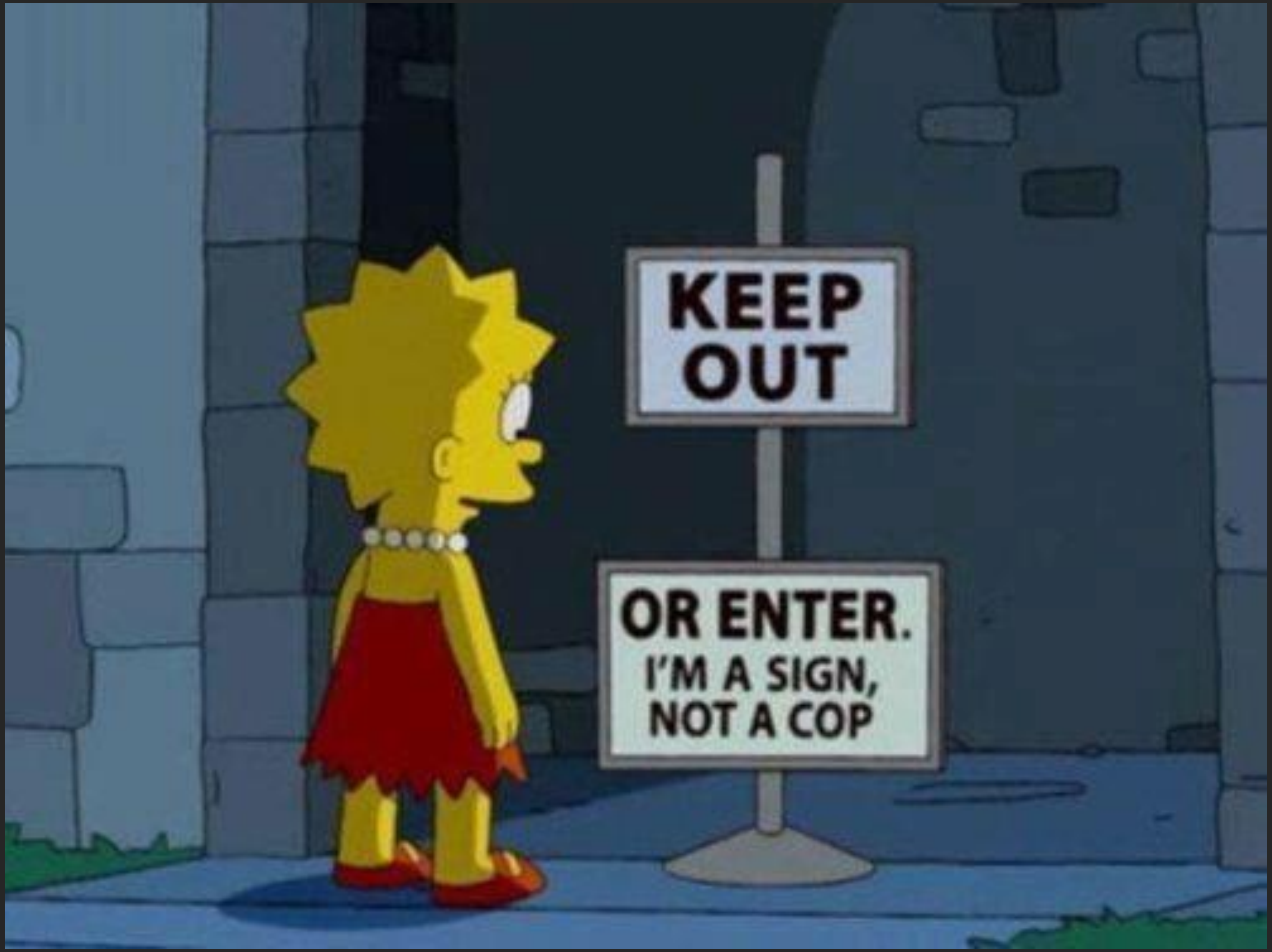
It’s 2019, but injection is still #1 in OWASP Top 10

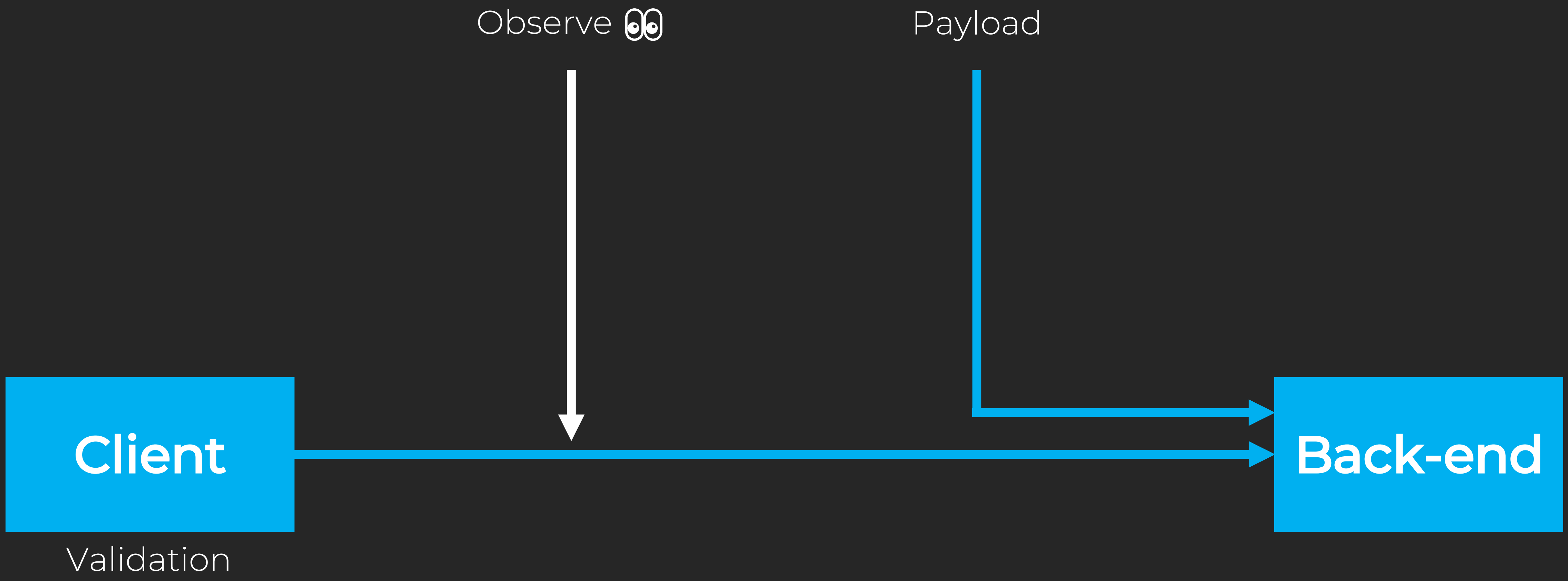
https://www.owasp.org/index.php/Top_10-2017_A1-Injection

Don’t trust data

https://news.ycombinator.com/item?id=8336025

Don’t just validate
client-side


KEEP
OUT
OR ENTER.
I'M A SIGN,
NOT A COP



Observe 👀

Payload

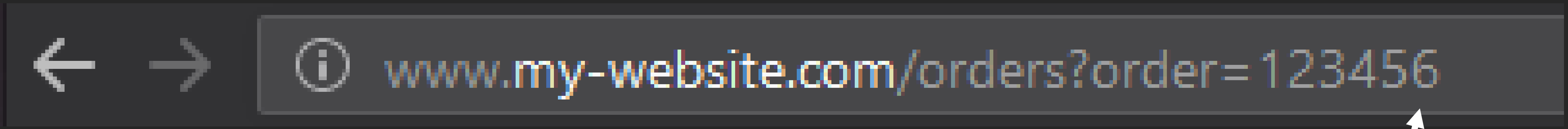
Client

Back-end

Validation

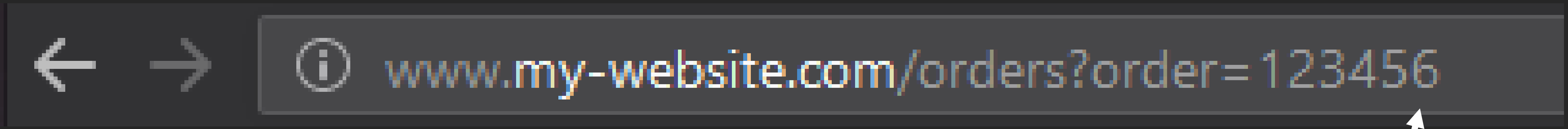
Broken access control

@Brunty



www.my-website.com/orders?order=123456

Do you trust this?



www.my-website.com/orders?order=123456

123457

?

https://en.wikipedia.org/wiki/Attribute-based_access_control

Don't trust users ~~input~~

@Brunty

Broken authentication

https://www.troyhunt.com/controlling-vehicle-features-of-nissan/

Hash passwords properly

Don’t use default passwords

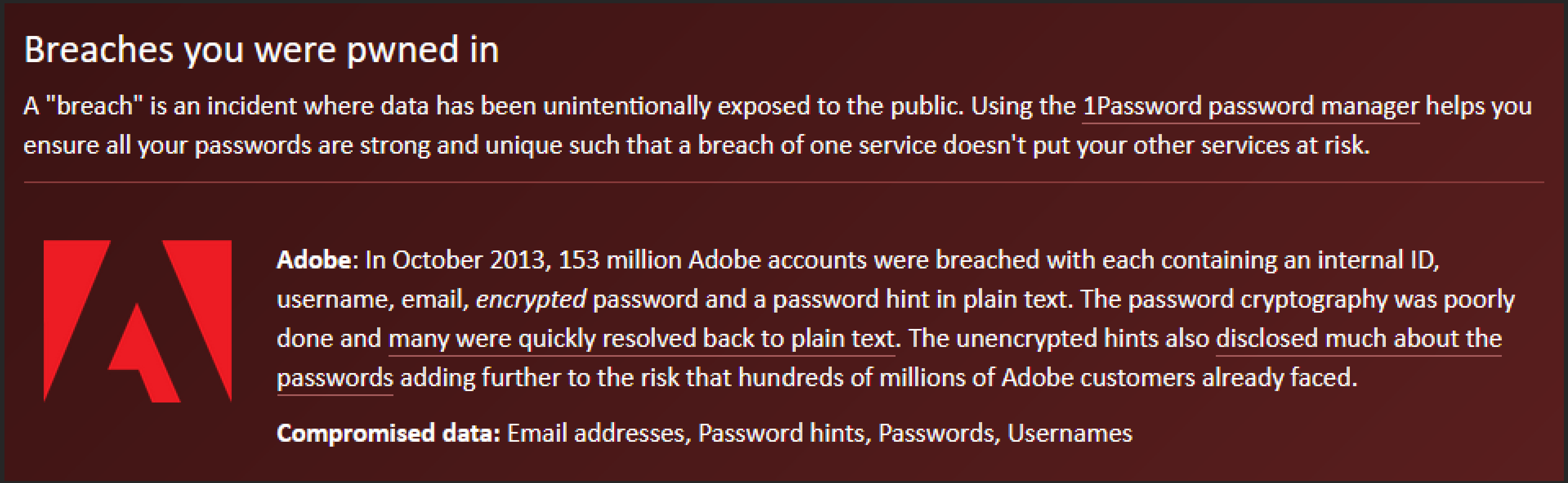
https://www.forbes.com/sites/kateoflahertyuk/2019/10/20/equifax-lawsuit-reveals-terrible-security-practices-at-time-of-2017-breach/

Don’t re-use passwords

https://blog.lastpass.com/2018/05/psychology-of-passwords-neglect-is-helping-hackers-win.html/

haveibeenpwned.com

@TroyHunt



Breaches you were pwned in

A "breach" is an incident where data has been unintentionally exposed to the public. Using the 1Password password manager helps you ensure all your passwords are strong and unique such that a breach of one service doesn't put your other services at risk.

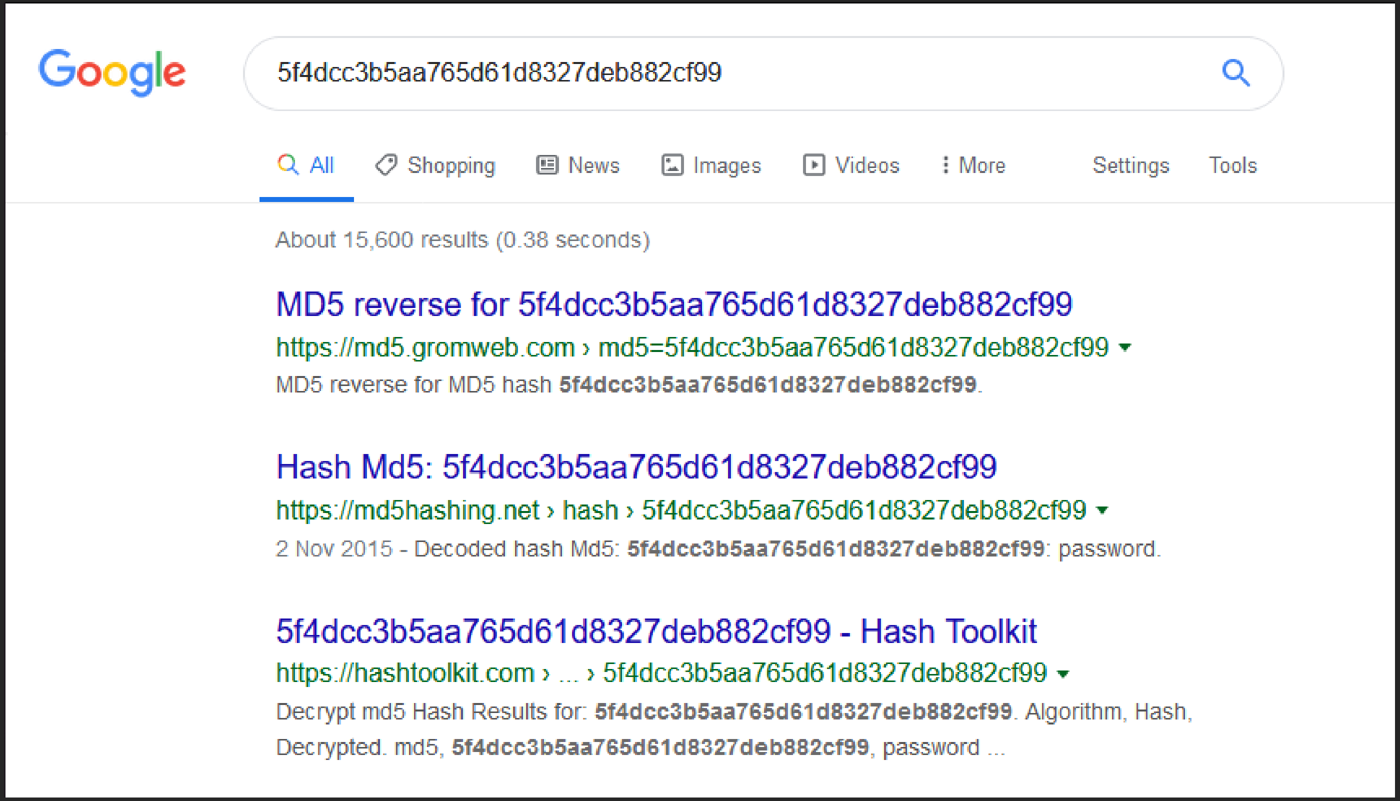
Adobe: In October 2013, 153 million Adobe accounts were breached with each containing an internal ID, username, email, *encrypted* password and a password hint in plain text. The password cryptography was poorly done and many were quickly resolved back to plain text. The unencrypted hints also disclosed much about the passwords adding further to the risk that hundreds of millions of Adobe customers already faced.

Compromised data: Email addresses, Password hints, Passwords, Usernames

Don’t allow your users to re-use passwords

@Brunty

5f4dcc3b5aa765d61d8327deb882cf99



About 15,600 results (0.38 seconds)

MD5 reverse for 5f4dcc3b5aa765d61d8327deb882cf99
https://md5.gromweb.com › md5=5f4dcc3b5aa765d61d8327deb882cf99 ▾
MD5 reverse for MD5 hash **5f4dcc3b5aa765d61d8327deb882cf99**.

Hash Md5: 5f4dcc3b5aa765d61d8327deb882cf99
https://md5hashing.net › hash › 5f4dcc3b5aa765d61d8327deb882cf99 ▾
2 Nov 2015 - Decoded hash Md5: **5f4dcc3b5aa765d61d8327deb882cf99**: password.

5f4dcc3b5aa765d61d8327deb882cf99 - Hash Toolkit
https://hashtoolkit.com › ... › 5f4dcc3b5aa765d61d8327deb882cf99 ▾
Decrypt md5 Hash Results for: **5f4dcc3b5aa765d61d8327deb882cf99**. Algorithm, Hash, Decrypted. md5, **5f4dcc3b5aa765d61d8327deb882cf99**, password ...

password

@Brunty

pwned passwords API

https://www.troyhunt.com/pwned-passwords-version-5/

Use

Multi Factor Authentication

But not SMS

What packages do you trust in your application?

https://help.github.com/en/articles/listing-the-packages-that-a-repository-depends-on#supported-languages

roave/security-advisories

https://github.com/Roave/SecurityAdvisories

```
php checker security:check /path/to/composer.lock
```

https://github.com/sensiolabs/security-checker

More packages than you think

Front-end

Mobile App(s)

Back-end

Platform / OS

Infrastructure

Keep them up-to-date

Death by a thousand paper-cuts

Mistakes will happen

Make sure you don’t miss the simple stuff

Mostly, it's not like the movies. (Sorry)

@Brunty

Expectation:

Reality:

Evaluate who you trust with data

Security at all stages of the project

Principle of least privilege

Encrypt data in transit and at rest

Check for public secrets

Don’t trust users & input

Hash passwords properly

Ensure your components aren’t vulnerable

OWASP Top Ten

Always be curious

Thanks!